Guiding Genetic Program Based Data Mining Using Fuzzy Rules

James F. Smith III and ThanhVu H. Nguyen

Code 5741
Naval Research Laboratory
Washington, DC, 20375-5320
jfsmith@drsews.nrl.navy.mil

Abstract. A data mining procedure for automatic determination of fuzzy decision tree structure using a genetic program is discussed. A genetic program (GP) is an algorithm that evolves other algorithms or mathematical expressions. Methods for accelerating convergence of the data mining procedure are examined. The methods include introducing fuzzy rules into the GP and a new innovation based on computer algebra. Experimental results related to using computer algebra are given. Comparisons between trees created using a genetic program and those constructed solely by interviewing experts are made. Connections to past GP based data mining procedures for evolving fuzzy decision trees are established. Finally, experimental methods that have been used to validate the data mining algorithm are discussed.

Keywords: Genetic Programs, Fuzzy Logic, Data Mining, Control Algorithms, Planning Algorithms.

1 Introduction

Two fuzzy logic based resource managers (RMs) have been developed that automatically allocate resources in real-time [1-3]. Both RMs were evolved by genetic programs (GPs). The GPs were used as data mining functions. Both RMs have been subjected to a significant number of verification experiments.

The most recently developed RM is the main subject of this paper. This RM automatically allocates unmanned aerial vehicles (UAVs) that will ultimately measure atmospheric properties in a cooperative fashion without human intervention [2,3]. This RM will be referred to as the UAVRM. It consists of a pre-mission planning algorithm and a real-time control algorithm that runs on each UAV during the mission allowing the UAVs to automatically cooperate.

The previous RM was evolved to control electronic attack functions distributed over many platforms [1]. It will be referred to as the electronic attack RM (EARM).

This paper introduces many novel features not found in the literature. These include several new approaches for improving the convergence of the genetic program that evolves control and planning logic. Such procedures involve the use of symbolic algebra techniques not previously explored, a terminal set that includes both fuzzy concepts and their complements, the use of fuzzy rules, etc. The control algorithm

E. Corchado et al. (Eds.): IDEAL 2006, LNCS 4224, pp. 1337 – 1345, 2006.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **Guiding Genetic Program Based Data Mining Using Fuzzy Rules**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Code 5741,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A data mining procedure for automatic determination of fuzzy decision tree structure using a genetic program is discussed. A genetic program (GP) is an algorithm that evolves other algorithms or mathematical expressions. Methods for accelerating convergence of the data mining procedure are examined. The methods include introducing fuzzy rules into the GP and a new innovation based on computer algebra. Experimental results related to using computer algebra are given. Comparisons between trees created using a genetic program and those constructed solely by interviewing experts are made. Connections to past GP based data mining procedures for evolving fuzzy decision trees are established. Finally, experimental methods that have been used to validate the data mining algorithm are discussed.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**9**	

evolved by a GP is compared to one created through expertise. Experiments to validate the evolved algorithm are discussed.

Section 2 gives a brief discussion of fuzzy decision trees (FDTs), how FDTs are used in the UAVRM, genetic programs and GP based data mining (DM). Section 3 describes the UAVRM's FDT that assign UAVs to paths. Section 4 examines how a fuzzy decision tree for the UAVRM was created through GP based data mining. Section 5 discusses experiments that have been conducted to validate the FDT that assigns UAVs to paths (AUP). Finally, section 6 provides a summary.

2 Fuzzy Decision Trees and Genetic Program Based Data Mining

The particular approach to fuzzy logic used by the UAVRM is the fuzzy decision tree [1-5]. The fuzzy decision tree is an extension of the classical artificial intelligence concept of decision trees. The nodes of the tree of degree one, the leaf nodes are labeled with what are referred to as root concepts. Nodes of degree greater than unity are labeled with composite concepts, i.e., concepts constructed from the root concepts [6,7] using logical connectives and modifiers. Each root concept has a fuzzy membership function assigned to it. Each root concept membership function has parameters to be determined. For the UAVRM, the parameters were set based on expertise.

The UAVRM consists of three fuzzy decision trees. Only the creation of the FDT by GP based data mining for assigning UAVs to paths will be considered in this paper. This FDT is referred to as the AUP tree; and the associated fuzzy concept, as AUP. The AUP tree makes use of the risk tree which is discussed in the literature [2, 3].

Data mining is the efficient extraction of valuable non-obvious information embedded in a large quantity of data [8]. Data mining consists of three steps: the construction of a database that represents truth; the calling of the data mining function to extract the valuable information, e.g., a clustering algorithm, neural net, genetic algorithm, genetic program, etc; and finally determining the value of the information extracted in the second step, this generally involves visualization.

In a previous paper a genetic algorithm (GA) was used as a data mining function to determine parameters for fuzzy membership functions [7]. Here, a different data mining function, a genetic program [9] is used. A genetic program is a problem independent method for automatically evolving computer programs or mathematical expressions.

The GP data mines fuzzy decision tree structure, i.e., how vertices and edges are connected and labeled in a fuzzy decision tree. The GP mines the information from a database consisting of scenarios.

3 UAV Path Assignment Algorithm, the AUP Tree

Knowledge of meteorological properties is fundamental to many decision processes. The UAVRM enables a team of UAVs to cooperate and support each other as they measure atmospheric meteorological properties in real-time. Each UAV has onboard its own fuzzy logic based real-time control algorithm. The control algorithm renders each UAV fully autonomous; no human intervention is necessary. The control algorithm aboard each UAV will allow it to determine its own course, change course to avoid danger, sample phenomena of interest that were not preplanned, and cooperate with other UAVs.

The UAVRM determines the minimum number of UAVs required for the sampling mission. It also determines which points are to be sampled and which UAVs will do the sampling. To do this, both in the planning and control stages it must solve an optimization problem to determine the various paths that must be flown. Once these paths are determined the UAVRM uses the AUP fuzzy decision tree to assign UAVs to the paths.

The AUP fuzzy decision tree is displayed in Figure 1. The various fuzzy root concepts make up the leaves of the tree, i.e., those vertices of degree one. The vertices of degree higher than one are composite concepts.

Starting from the bottom left of Figure 1 and moving to the right, the fuzzy concepts "risk-tol," "value", "fast," and "low risk," are encountered. These concepts are developed in greater mathematical detail in the literature [2,3]. The fuzzy concept "risk-tol" refers to an individual UAV's risk tolerance. This is a number assigned by an expert indicating the degree of risk the UAV may tolerate. A low value near zero implies little risk tolerance, whereas, a high value near one implies the UAV can be subjected to significant risk.

The concept "value" is a number between zero and one indicating the relative value of a UAV as measured against the other UAVs flying the mission. The concept "value" changes from mission to mission depending on which UAVs are flying.

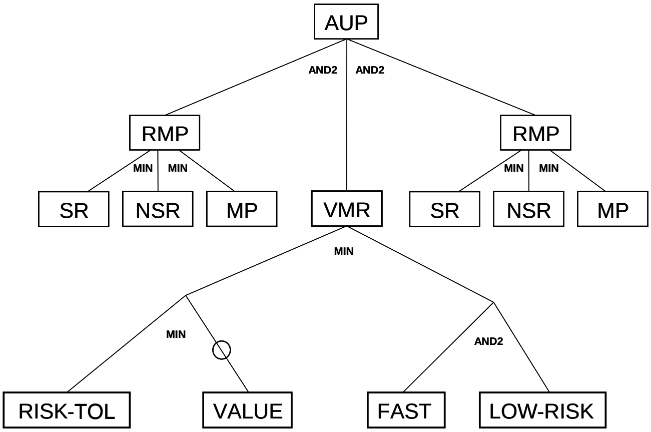

Fig. 1. The AUP subtree for the UAVRM

The concept "fast" relates to how fast the UAV is and builds in measures of the UAV's reliability estimates as well as its risk tolerance and the mission's priority.

The rightmost concept is "low risk." It quantifies experts' opinions about how risky the mission is. It takes a value of one for low risk missions and a value near zero for high risk missions.

These four fuzzy root concepts are combined through logical connectives to give the composite concept "VMR." Although four concepts are now used to construct

VMR it originally only used the concepts related to value and mission risk, and was called the Value-Mission-Risk (VMR) subtree.

Each vertex of the "VMR" tree uses a form of "AND" as a logical connective. In fuzzy logic, logical connectives can have more than one mathematical form. Based on expertise it was useful to allow two types of ANDs to be used. The two mathematical forms of AND used are the "min" operator and the algebraic product denoted in Figure 1 as "AND2." When a "min" appears on a vertex then the resulting composite concept arises from taking the minimum between the two root concepts connected by the "min." When an "AND2" appears it means that the resulting composite concept is the product of the fuzzy membership functions for the two concepts connected by the AND2.

The final subtree of AUP that needs to be described is the reliability-mission priority (RMP) subtree. The RMP tree appears twice on the AUP tree. RMP consists of a "min" operation between three fuzzy concepts. These concepts are "sr" which refers to an expert's estimate of the sensor reliability, "nsr" which refers to an expert's estimate of the non-sensor system reliability and "MP" a fuzzy concept expressing the mission's priority.

The AUP tree is observed to consist of the VMR subtree and two copies of the RMP subtree with AND2 logical connectives at each vertex. These fuzzy concepts and their related fuzzy membership functions, as well as additional details are given in much greater detail in [2, 3].

The AUP tree given in Figure 1 was originally created using human expertise alone. The rediscovery of this tree using GP based data mining is described in the next section.

4 GP Creation of the AUP Tree

The terminal set, function set, and fitness functions necessary for the GP to be used as a data mining function to automatically create the AUP tree are described below. The terminal set used to evolve the AUP tree consisted of the root concepts from the AUP tree and their complements. The terminal set, T, is given by

$$\begin{gathered}\text{T=\{risk-tol, value, fast, low-risk, sr, nsr, MP, not-risk-tol, not-valuable,}\\ \text{not-fast, not-low-risk, not-sr, not-nsr, not-MP\}.}\end{gathered} \tag{1}$$

Let the corresponding fuzzy membership functions be denoted as

$$\begin{gathered}\{\mu_{risk-tol},\mu_{value},\mu_{fast},\mu_{low-risk},\mu_{sr},\mu_{nsr},\dots\\ \mu_{MP},\mu_{not-risk-tol},\mu_{not-valuable},\mu_{not-fast},\dots\\ \mu_{not-low-risk},\mu_{not-sr},\mu_{not-nsr},\mu_{not-MP}\}.\end{gathered} \tag{2}$$

When mathematical expressions are constructed by a GP that reproduce the entries in a database within some tolerance, the process is referred to as symbolic regression [10]. It is found in symbolic regression that candidate solutions are frequently not in algebraic simplest form and this is the major source of their excess length. When candidate solutions are too long this is referred to as bloat [10].

By including in the terminal set a terminal and its complement, e.g., “risk-tol,” and “not-risk-tol”; “value” and “not-valuable”; etc., it is found that bloat is less and convergence of the GP is accelerated. This is a recent innovation which was not used when the EARM was evolved using GP based data mining (DM) [1]. Additional bloat control procedures are described below.

The mathematical form of the complement whether it appears in the terminal set or is prefixed with a “NOT” logical modifier from the function set is one minus the membership function. To make this more explicit

$$\mu_{NOT(A)} = \mu_{not-A} = 1 - \mu_A, \tag{3}$$

where *NOT(A)* refers to the application of the logical modifier *NOT* from the function set to the fuzzy concept *A* from the terminal set. The notation, *not-A* refers to the terminal which is the complement of the terminal *A*.

The function set, denoted as F, consists of

$$F=\{AND1, OR1, AND2, OR2, NOT\}, \tag{4}$$

where the elements of (4) are defined in (5-9). Let A and B represent fuzzy membership functions then elements of the function set are defined as

$$AND1(A,B) = min(A,B); \tag{5}$$

$$OR1(A,B) = max(A,B); \tag{6}$$

$$AND2(A,B) = A \cdot B; \tag{7}$$

$$OR2(A,B) = A + B - A \cdot B; \tag{8}$$

and

$$NOT(A) = 1 - A. \tag{9}$$

The database to be data mined is a scenario database kindred to the scenario database used for evolving the EARM [1]. In this instance scenarios are characterized by values of the fuzzy membership functions for the elements of the terminal set plus a number from zero to one indicating the experts’ opinion about the value of the fuzzy membership function for AUP for that scenario.

GPs require a fitness function [9]. As its name implies the fitness function measures the merit or fitness of each candidate solution represented as a chromosome. The fitness used for data mining is referred to as the input-output fitness.

The input-output fitness for mining the scenario database takes the form

$$f_{IO}(i, n_{db}) \equiv \frac{1}{1 + 2 \cdot \sum_{j=1}^{n_{db}} \left| \mu_{gp}(i, e_j) - \mu_{\mathrm{exp}ert}(e_j) \right|}. \tag{10}$$

where e_j is the j^{th} element of the database; n_{db} is the number of elements in the database; $\mu_{gp}(e_j)$ is the output of the fuzzy decision tree created by the GP for the i^{th} element of the population for database element e_j; and $\mu_{expert}(e_j)$ is an expert’s estimate as to what the fuzzy decision tree should yield as output for database element e_j.

The AUP tree is evolved in three steps. The first step involves evolving the VMR subtree; the second step, the RMP subtree and the final step, the full AUP tree. In the second and third steps, i.e., evolving the RMP subtree and full AUP tree from the RMP and VMR subtrees, only the input-output (IO) fitness in (10) is calculated, i.e., the rule-fitness described below is not used.

When evolving the VMR subtree a rule-fitness is calculated for each candidate solution. Only when the candidate's rule fitness is sufficiently high is its input-output fitness calculated. The use of the rule-fitness helps guide the GP toward a solution that will be consistent with expert rules. Also the use of the rule fitness reduces the number of times the IO fitness is calculated reducing the run time of the GP. After some preliminary definitions of crisp and fuzzy relations, a set of crisp and fuzzy rules that were used to help accelerate the GP's creation of the VMR subtree are given. The rules are combined to formulate the rule fitness. The mathematical form of the rule fitness has not been included due to space limitations.

Let T be a fuzzy decision tree that represents a version of the VMR subtree, that is to be evolved by a genetic program. Let A and B be fuzzy concepts. Then let $\gamma_{share}(T,A,B)=1$ if A and B share a logical connective denoted as C and $\gamma_{share}(T,A,B)=0$, otherwise.

Furthermore, define the fuzzy relation

$$\mu_{com}(T,A,B,C)=\begin{cases} 0.4 \quad if \quad C=AND1 \quad or \quad AND2 \\ 0.1 \quad if \quad C=OR1 \quad or \quad OR2 \\ 0, \quad otherwise \end{cases} \cdot \tag{11}$$

The following is a subset of the rules used to accelerate the GP's convergence and to help produce a result consistent with human expertise.

R1. "not-valuable" and "risk-tol" must share a logical connective, denoted as C_1, i.e., it is desired that $\gamma_{share}(T,not-valuable,risk-tol)=1$

R2. "not-valuable" and "risk-tol" strongly influence each other, so they should be connected by AND1 or AND2. So it is desired that $\mu_{com}(T,not-valuable,risk-tol,C_1)=.4$

R3. "fast" and "low-risk" have an affinity for each other. They should share a logical connective, denoted as C_2, i.e., it is desired that $\gamma_{share}(T,fast,low-risk)=1$

R4. The fuzzy root concepts "fast" and "low-risk" strongly influence each other, so they should be connected by AND1 or AND2. So it is desired that $\mu_{com}(T,fast,low-risk,C_2)=.4$.

R5. There is an affinity between the fuzzy root concepts $C_1(not-valuable,risk-tol)$ and $C_2(fast,low-risk)$, they are connected by a logical connective denoted as C_3, i.e., it is desired that,

$$\gamma_{share}(T,C_1(not-valuable,risk-tol),C_2(fast,low-risk))=1. \tag{12}$$

When the EARM was evolved by GP based data mining [1] bloat was controlled using adhoc procedures based on tree depth and parsimony pressure. Most of the bloat in evolving mathematical expressions with a GP arises from the expressions not being in algebraic simplest form [10]. With that observation in mind, computer algebra routines have been introduced that allow the GP to simplify expressions. The following is a partial list of algebraic simplification techniques used during the evolution of the EARM and the AUP tree. The simplification routines used when evolving AUP are more sophisticated than those applied to the creation of EARM [1].

One routine simplifies expressions of the form N*OT(NOT(A)) = A.* This can be more complicated than it initially appears, since the NOT logical modifiers can be separated on the fuzzy decision tree.

Another simplification procedure consists of eliminating redundant terminals connected by an AND1 logical connective. An example of this is *AND1(A,A) =A.* Like the case with the logical modifier NOT there can be a separation between the AND1s and the terminals that add complexity to the simplification operation.

The third algebraic simplification example is like the second. It involves simplifying terminals connected by OR1s. Like AND1, separation between terminals and OR1 can increase the complexity of the operation.

Other types of algebraic simplification use DeMorgan's theorems in combination with the above procedures. This can significantly reduce the length of an expression.

Another algebraic procedure that reduces the length of expressions includes replacement of forms like AND2(A,A) by the square of "A," i.e., A^2. Still another length reducing simplification includes replacing NOT(A) with not-A, its complement from the terminal set listed in (1).

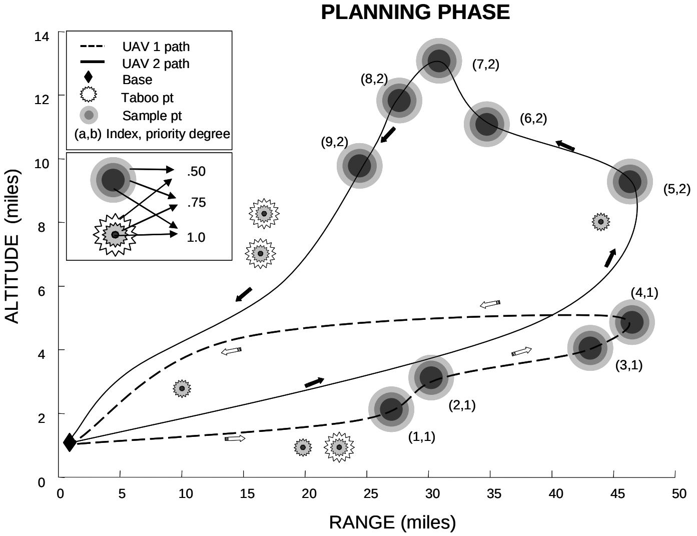

Fig. 2. Trajectory of two UAVs as determined by the planning algorithm and their paths assigned by AUP

There is always a question of how much algebraic simplification should be conducted from generation to generation as such the simplification algorithm allows levels of simplification. If a low level of simplification is selected then some parts of an expression remain that might be eliminated during full simplification. This has two advantages: it leaves chromosome subcomponents that may prove useful during mutation or crossover and it takes less CPU time.

Algebraic simplification produces candidate solutions in simpler form making it easier for human observers to understand what is being evolved. Having candidate solutions that are easier to understand can be an important feature for improving the evolution of GPs.

5 Computational Experiments

The AUP tree described above has been the subject of a large number of experiments. This section provides a description of an experiment that is representative of the type of scenarios designed to test the AUP tree. Due to space limitations only an experiment involving two UAVs is discussed.

In Figure 2 a scenario using two UAVs illustrates how AUP properly assigns the UAVs to the best path. The two paths were created by the planning algorithm so that the UAV could most efficiently sample the atmosphere's electromagnetic index of refraction [2, 3].

Sample points are labeled by concentric circular regions colored in different shades of gray. The lighter the shade of gray used to color a point, the lower the point's grade of membership in the fuzzy concept "desirable neighborhood." [2, 3] The legend provides numerical values for the fuzzy grade of membership in the fuzzy concept "desirable neighborhoods." If the fuzzy degree of desirability is high then the index of refraction is considered to be close to the index of refraction of the sample point at the center of the desirable neighborhood. This allows the UAV to make significant measurements while avoiding undesirable neighborhoods.

Each sample point is labeled with an ordered pair. The first member of the ordered pair provides the index of the sample point. The second member of the ordered pair provides the point's priority. For example, if there are n_{sp} sample points and the q^{th} sample point is of priority p, then that point will be labeled with the ordered pair (q,p).

Points surrounded by star-shaped neighborhoods varying from dark grey to white in color are taboo points. As with the sample points, neighborhoods with darker shades of gray have a higher grade of membership in the fuzzy concept "undesirable neighborhood." The legend provides numerical values for the fuzzy grade of membership in the fuzzy concept "undesirable neighborhood." UAVs with high risk tolerance may fly through darker grey regions than those with low risk tolerance.

UAVs start their mission at the UAV base which is labeled with a diamond-shaped marker. They fly in the direction of the arrows labeling the various curves in Figure 2.

Figure 2 depicts the sampling path determined by the planning algorithm for an experiment involving two UAVs. The first, UAV(1) follows the dashed curve; the second, UAV(2), the solid curve. The UAVs were assigned to the different paths by the

AUP fuzzy decision tree described in section 2. UAV(1) is assigned to sample all the highest priority points, i.e., the priority one points. UAV(2) samples the lower priority points, i.e.; those with priority two. Due to the greedy nature of the point-path assignment algorithm, the highest priority points are assigned for sampling first.

6 Summary

A genetic program (GP) has been used as a data mining (DM) function to automatically create decision logic for two different resource managers (RMs). The most recent of the RMs, referred to as the UAVRM is the topic of this paper. It automatically controls a group of unmanned aerial vehicles (UAVs) that are cooperatively making atmospheric measurements.

The DM procedure that uses a GP as a data mining function to create a subtree of UAVRM is discussed. The resulting decision logic for the RMs is rendered in the form of fuzzy decision trees. The fitness function, bloat control methods, data base, etc., for the tree to be evolved are described. Innovative bloat control methods using computer algebra based simplification are given. A subset of the fuzzy rules used by the GP to help accelerate convergence of the GP and improve the quality of the results is provided. Experimental methods of validating the evolved decision logic are discussed to support the effectiveness of the data mined results.

References

1. Smith, III, J. F.: Fuzzy logic resource manager: decision tree topology, combined admissible regions and the self-morphing property, In: Kadar, I. (ed.): Signal Processing, Sensor Fusion, and Target Recognition XII: Vol. 5096, SPIE Proceedings, Orlando (2003) 104-114.
2. Smith, III, J. F., Nguyen, T., H.: Distributed autonomous systems: resource management, planning, and control algorithms, In: Kadar, I. (ed.): Signal Processing, Sensor Fusion, and Target Recognition XIV: Vol. 5096, SPIE Proceedings, Orlando (2005) 65-76.
3. Smith, III, J. F., Nguyen, T., H.: Resource manager for an autonomous coordinated team of UAVs, In: Kadar, I. (ed.): Signal Processing, Sensor Fusion, and Target Recognition XV: 62350C, SPIE Proceedings, Orlando (2006) 104-114.
4. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House, Boston (1999)
5. Tsoukalas, L.H., Uhrig, R.E.: Fuzzy and Neural Approaches in Engineering: John Wiley and Sons, New York (1997)
6. Zimmerman, H. J.: Fuzzy Set Theory and its Applications. Kluwer Academic Publishers Group, Boston (1991)
7. Smith, III, J.F., Rhyne, II, R.: A Resource Manager for Distributed Resources: Fuzzy Decision Trees and Genetic Optimization. In: Arabnia, H. (ed.): Proceeding of the International Conference on Artificial Intelligence, IC-AI'99, Vol. II. CSREA Press, Las Vegas (1999) 669-675.
8. Bigus, J.P.: Data Mining with Neural Nets. McGraw-Hill, New York, (1996).
9. Koza, J.R., Bennett III, F.H.: Andre, D., Keane, M.A.: Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann Publishers, San Francisco (1999).
10. Luke, S., Panait, L.: Fighting Bloat with Nonparametric Parsimony Pressure. In: Guervos, J.J.M (ed.): Parallel Problem Solving from Nature - PPSN VII, 7th International Conference. Proceedings. LNCS Vol. 2439, Springer-Verlag, Berlin, (2002) 411-421.